

LESSONS LEARNED FROM THE SNS RELATIONAL DATABASE*

D. Purcell, J. Patton, E. Danilova, ORNL, Oak Ridge, TN 37830, USA

Abstract

The Spallation Neutron Source Project relies heavily on many different applications that require and depend on the SNS integrated relational database. Although many of the projects undertaken have been successful, the majority of time and energy spent on producing products has resulted in opportunities lost. The percentage of time lost or wasted has been very similar to that of software development projects everywhere. At the SNS the variety of factors that have influenced these projects can be traced to some specific areas: management support, project deadlines, user expectations, graphical user interfaces, and the database itself. This paper presents a look at the factors that have helped make some projects a success and the factors that have led to less favorable results.

OVERVIEW OF SNS RELATIONAL DATABASE

The Spallation Neutron Source (SNS) has deployed a global integrated ORACLE-based relational database (RDB). The RDB was developed to support many different aspects of the SNS project. This includes but is not limited to data structures that support project administration, equipment installation, SNS operations, project documentation and the SNS control system. Because the RDB spans many functions of the SNS project, it has become the central storage area for a vast amount of data and it is considered by most to be the main source for information and support data.

Some working with the control system have tried to take advantage of this. The control system "area" within the SNS RDB is the most developed. Many different types of data have been captured. The RDB contains beam line equipment support data, networking data, installation data, calibration data, machine setup data, machine protection system (MPS) data and input output controller (IOC) data.

RDB Strategy

From the very beginning, it was the goal of some in management to use a relational database in support of SNS activities. Included in these goals were the desire to foster communications, and to provide quality and configuration control. Past experience had given engineers and management a taste of the benefits a RDB could provide. So, in support of the activities of SNS, the RDB would be a tool that was used whenever it was beneficial.

RDB Evolution

The growth of the database schema was dictated by the growth of SNS. First, structure was produced to support

the business aspects of the project. Then support was expanded to procurement and tracking of equipment. Finally, the schema was created to handle installation and operation of SNS. This growth was not coordinated. The RDB was controlled by the SNS database administrator (DBA) to the extent that duplication of data wasn't allowed and that the database remained normalized. The result was a well-formed RDB that is capable of handling a large amount of data.

During this evolution, management and software engineers have always used the RDB as a tool that supports the efforts of SNS staff. But the RDB tool is used only when practical and its use was and continues to be a secondary effort. The primary goal at SNS will always be "beam on target".

SUCCESS OR NOT

The word "failure" as used throughout this paper indicates lost possibility to expand the use of the database. It does not mean that a specific project has failed or that the goal of the software writers was not accomplished.

The overall success or failure of the SNS RDB has not been determined yet. We believe that we have already been successful because of the continued use of the database in one way or another. Many users are new to the use of an RDB. There are still some that believe RDB support is not beneficial to their goal. But we have seen that these individuals can change and become advocates for the use of the RDB. One of our leading indicators of success is that individuals that leave SNS now wish to continue implementing RDB support in their new endeavors. Success is the continued development of ideas on how we can use the database in the future.

CHARACTERISTICS OF SUCCESSFUL SNS RDB PROJECT

Many RDB projects have been started at SNS. Some of these projects have been completed successfully. But a majority of them have never been finished or have been created and not put to use. When we look at the applications that we consider a success, we find that there are common characteristics.

Project Champion

A project champion might be the most important factor in whether a RDB project succeeds or fails. At SNS, few individuals have had a desire to start and stick with an RDB project. So, ideas and efforts supported by individuals quickly become the projects that are supported heavily by database developers. Loss of a champion from

* Oak Ridge National Laboratory is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy.

a specific project, however, usually means the end of that effort.

Historic Reference

Developers that have produced software before that incorporated a RDB then they will likely do so again. However, if these projects grow to be unmanageable because of complexities of the SNS database requirements (as in cumbersome schemas or data needs from others), we have seen that the developer will likely use their own RDB and that the data will not be linked to the SNS RDB.

Also, individuals that are without a historic perspective in a situation will use what is given to them. So, if working RDB applications are given to new employees or employees in new situations, then those applications will continue to be needed and used.

Real Need and Code Stealing

A sure way to get a project completed is to absolutely require it. If a tool is needed and the only way to do it is via a RDB then we have been successful.

We have also found that applications that can be duplicated from something else and then “tweaked” to accomplish a new objective have led to successful completion. It should be noted that this has also had negative consequences because there have sometimes been additional requirements on the new project only because of the borrowed code. This, too, can lead to failure.

INFLUENCES ON RDB PROJECT FAILURE

As stated, many different projects have been started with the database involved and a majority of these projects have led to frustration and an unfavorable result. While trying to define what makes a project successful, we also noted what has caused the opposite.

Deadlines

Because of the relentless “beam on target” goal, many engineers started with the goal of database integration but would later have to abandon it. The deadlines required for commissioning would lead software engineers to not use the SNS RDB because those engineers could accomplish goals faster using other means. This is not to say that they didn’t use relational databases. There are many instances throughout SNS where databases exist on a desktop PC in support of some effort.

Deadlines also influenced those doing work with hardware. Hardware engineers and technicians don’t want to deal with the effort involved in getting information about their work into a database. The lack of easy to use tools and organization on how to input this data made database use difficult. And deadlines give employees an easy excuse to not do it.

User/Client Expectations

Some client expectations aren’t realistic. Often this is a problem of perception: users tend to think locally and short term rather the globally and long term. They then miss out on many opportunities to use software systems to aid them in their tasks. This is where management support becomes critical. With many projects at SNS, the clients expected fully functioning applications but were not prepared to go through the process of developing the database or other tools that were needed to get started.

Data In Versus Out

Data should be entered and extracted in a simple manner. At SNS, some might argue that this is the major reason for the lack of use of the RDB. User interfaces are awkward and not a standardized. It is this very diversity that has hurt because tools are not polished or fully completed. The lack of a standard toolset causes frustration in potential users which causes them to abandon the SNS RDB approach.

Complex Database Structure

Some would argue that the complexity of the SNS database schema has affected the use of the RDB in both positive and negative ways. The negative is that data that might be easy to get at without the complex schema is sometimes burdened with requirements that aren’t necessary for the specific project. Relationships and conventions are built in to the schemas that sometimes require more effort to support than a user, manager, or database expert is willing to invest.

However this very complexity has given SNS the ability to track various relationships and report on them. Many mechanisms are available to hide the underlying complexity from an end user. This has allowed users to see related data that otherwise might be hard to get at.

Management Support

Any software deployment strategy requires strong management support. At SNS, all performance evaluations were based on completion of construction and beam on target. This pushed software designers to do their work however they could to get the job done. If it meant slowing down their specific goal, software developers and those working with hardware were allowed to ignore requests involving the database.

ATTITUDE

Everyone at SNS has a different opinion of the RDB.

Database Developers

Relational databases have the inherent ability to give a great deal of information. Those that work with the SNS RDB also understand that the database is only useful when it is populated with accurate data. The complexity of the SNS RDB is viewed as an opportunity to expand its usefulness. As the SNS project moved through

construction the deviations from a standard use of the RDB have been seen as a lost opportunity. This lost opportunity was a frustration in the beginning but is now seen apparently accepted at SNS.

Software Engineers

SNS employs a wide variety of individuals that provide software. Most of these people are experienced enough to understand the potential of the RDB. However, without a history of interaction between their applications and a relational database, their desire to change their proven technique is very low. The possibility of use does increase with a historical use of a RDB but complexities incorporated by the SNS RDB can still be a deterrent. In general, software writers view the modification of their applications to include use of the SNS RDB as unnecessary. The lack of a managerial mandate and the pressure of impending deadlines usually mean that it is impossible to get an application written which takes advantage of an RDB. This does not mean that software writers are against the RDB. The RDB complexity does cause frustration but overall the RDB is considered a potential tool that can be used to get a job done. Use of the tool, however, remains low.

Management

At SNS, most of the management believes use of the SNS RDB is a good idea. Some are indifferent. Just as individual software writers view the RDB as a tool, so too does management. Most have made decisions concerning a groups' direction based on ability to complete their work. They have not enforced what would be considered a constraint or something that might hinder their employee's ability to produce results. And because there hasn't been anything done with the RDB that is ground breaking, most managers are apathetic in regards to the use of the RDB.

General Users

Most potential users of the database now believe that it is finished, that the dataset is populated, and that the tools available are finished. Because none of this is true, most SNS users are still frustrated when it comes to using the RDB. The perception is that the effort should not be in entering data or building an interface and that now the RDB should be paying dividends. The general user does, at times, recognize a benefit and this has changed that person into a "Project Champion". But the usual outcome is for potential users to look at what is available and decide at that point if it is useful. Typically, the application or tool isn't exactly what they need or want so the user will move to a tool that provides them more comfort.

CONCLUSION

SNS

Use of the SNS RDB schema and all that it includes: naming conventions, equipment and non-control system data, etc., seems to be the best approach to allow growth. Users of the database will come from different areas. SNS projects that use the database will be ones that are created as part of some specific task and the complexity that the SNS schema has gives us the ability to handle most jobs. Without the current schema, RDB jobs would be a patchwork of tables and interfaces.

The use of the SNS RDB can not be mandated by management and shouldn't be. There isn't enough staff to handle all of the potential work. Procedures and standardization should have been enforced from the start of SNS. But for future work that does involve the RDB, SNS should implement some sort of standardization that is supported by management-backed procedures. Forcing the use of these procedures would eventually give SNS a standard toolset and dataset that users would trust.

Summary

The SNS RDB has had a design and development period where mistakes were made and shortcuts were taken. This none the less has given us a strong base to build upon. The lost opportunities were frustrating but proved to be blocks from which we could continue to grow.

Now, we have many successful implementations of software using the RDB. Although there is still frustration, these successes have led management to start a few projects that mandate the use of the procedures which include the use of the database. The successes have also given other developers some better examples of what is possible.

However until the RDB is used commonly, the growth of users and functionality is contingent on the outlook of the individuals working on the different tasks. Database developers need to continue to find project champions and to produce standardized user tools that will interest the software writers and general users. Future users must see existing toolsets as potential ways of benefiting their work. It is up to the individual users to continue to incorporate the use of the RDB.

REFERENCES

- [1] J. David Purcell, John Galambos and Jeff Patton, "Initial Experience with Database Applications", ICALEPCS 2003, Gyeongju, Korea, October 2003, <http://www.jacow.org>.